170602 逆向-CrackMe之009

原创

奈沙夜影 于 2017-06-02 18:56:36 发布 311 收藏
分类专栏： CrackMe

本文链接：https://blog.csdn.net/whklhhhh/article/details/72847045


CrackMe 专栏收录该内容
83 篇文章 2 订阅
订阅专栏

1625-5 王子昂 总结《2017年6月2日》【连续第244天总结】

A. CrackMe(9)

B. 今天的是一个Name/Serial类型的，拖入PEiD显示VB-无壳

随便输入错误以后弹窗，虽然内容看不懂不过肯定是错误就对了

在OD中暂停，ALT+K显示堆栈，找到MsgBox的调用，跟过去

向上翻一点看到了错误窗口的字符串，再往上就看到了另外一段不明意义的字符串

再向上就是一个test和je跳转，NOP之，爆破成功

算法部分我完全卡死了，从头跟下去但是无论寄存器还是栈堆中都只出现了两次字符串，其他没有任何计算结果

整个流程中出现了大量API，结合之前提到过的VB常见函数能大概理解意思

但是VB与Delphi有一个很大的不同，它的函数返回值不放在寄存器中，而是在之前push进去的堆栈之中。而根据push的参数却又

中间算法流程比较清晰：进行循环，提取Name的每个字符，转换成Unicode后再转换成ASCII值

但是接下来调用了一个Add的API就让人看不懂了，它push的参数与之前的ASCII不是同一个地址，加到最后总是02

然后又用2乘以3，最后结果的地方出现的是5

最后的比较API也不理解，无论输入什么都是03和08在比较

没办法，看论坛查writeup

原来VB的变量保存形式是：前两个4字节保存的是数据类型之类的相关类型，第三个4字节才是真正的字符串

所以我盯的位置都是第一个4字节，当然没有有意义的内容了

知道了这点再去重新跟一遍

提取Name的每个字符，转换成ASCII值后累加，最后再乘以0x499602D2（其实就是十六进制的1234567890）

得到的结果通过vbaMidStmtVar来将第四个和第九个字符换成'-'，最后直接与Serial比较即可

与Delphi相比，VB的过程大量的使用API，包括rtcMidCharVar取出字符， rtcAnsiValueBstr将Unicode转换成ASCII，__vbaVarAc

还有一些freeObj,freeVarList等等是释放作用的，就不用管了

要经常关注堆栈（的下两个字节）而不是寄存器_(:з」∠)_麻烦许多

C. 明日计划

CrackMe(10)